

提出日：2024 年 4 月 26 日

先進情報プロジェクト実習

テーマ IT・1 第 2 回レポート

学籍番号：C0A22113

氏名：成田彩華

設定ファイルの確認

設定ファイルの確認を行なっていく。以下のコマンドを入力する。

```
$ sudo apache2ctl configtest
```

以下のようなエラーが出るかを確認する。出なければ設定ができています。

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain
name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this
message

Syntax OK
```

このようなエラーが出た場合には、以下のコマンドを入力し、設定ファイルを root 権限で作成する。

```
sudo touch /etc/apache2/conf-available/fqdn.conf
```

このファイルの内容は次の通りである。

```
ServerName myserver
```

以下のコマンドで、この設定を有効にして Apache2 を再起動する。

```
$ sudo a2enconf fqdn

$ sudo systemctl reload apache2
```

先ほどのコマンドでエラーが出ないか再度確認する。

```
$ sudo apache2ctl configtest

Syntax OK
```

上記のように Syntax OK のみ表示されれば問題なく設定ができています。

PHP のインストール

PHP のインストールを進めていく。以下のコマンドを入力し、インストールしていく。

```
$ sudo apt install php libapache2-mod-php
```

この操作後に追加で21.3 MB のディスク容量が消費されます。

```
続行しますか? [Y/n]
```

このように聞かれたら Y を入力して続行する。すると PHP がインストールされる。

PHP のバージョンを確認し、インストールされているか確認する。

```
$ php -v

PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)

Copyright (c) The PHP Group

Zend Engine v4.1.2, Copyright (c) Zend Technologies

with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
```

このようにバージョンが表示されているため、問題なくインストールがされている。

PHP の動作確認

PHP の動作確認を行なっていく。/var/www/html/test に以下の内容の info.php を作成する。

```
<?php
    phpinfo();
?>
```

Ubuntu の IP アドレスを確認する。

```
$ hostname -I
192.168.64.7
```

ブラウザから Ubuntu の Web サーバにある info.php にアクセスする。

<http://192.168.64.7/test/info.php> をウェブサーバーに入力するとアクセスできる。

「PHP Version 8.1.2-1ubuntu2.14」のように書かれている表が表示されれば、PHP は動作していることが確認できる。

接続できたことが確認できたため、セキュリティのため、パーミッションを変更しておく。Web サイトからアクセスするユーザは www-data であるため、所有者を www-data にしておく。また、このファイルは所有者が読み込みできれば良いため、以下のコマンドで、読み込み権限だけ与えておく設定に変えておく。

```
$ sudo chown www-data:www-data /var/www/html/test/info.php
$ sudo chmod 400 /var/www/html/test/info.php
```

再度、Web ブラウザから info.php にアクセスし、パーミッション変更前と同じ画面になれば、パーミッション変更が正常に行えている。

SQLite3 のインストール

SQLite3 のインストールを行なっていく。次のコマンドでインストールする。

```
$ sudo apt install sqlite3
```

SQLite の起動していく。以下のコマンドを入力して起動する。

```
$ sqlite3
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

このように sqlite>というプロンプトに入力する状態になり、起動が完了する。この除隊で SQLite が扱える。

ヘルプは以下のコマンドで表示する。

```
sqlite> .help
```

コマンドの説明が表示される。

使用するデータベースで設定されている文字コードについては、次のコマンドで確認できる。UTF-8 になっているはずなので特に問題はない。

```
sqlite> PRAGMA encoding;
```

```
UTF-8
```

SQLite を終了する時は次のコマンドで行う。

```
sqlite> .exit
```

または、

```
sqlite> .quit
```

CTRL + c では終了できないので注意する。上記のコマンドを入力すると元の Linux のターミナルのプロンプトに戻る。

SQLite3 のデータベース作成

データベースの作成を行なっていく。まずは、以下のコマンドで SQLite を起動する。

```
$ sqlite
```

新しいデータベースファイルを作成する。次のコマンドでは、新規作成でデータベースをオープンしている。

```
sqlite> .open -new /var/www/html/test/test.db
```

2 回目以降はデータベースを指定してオープンすれば良い。

```
sqlite> .open /var/www/html/test/test.db
```

ここでは、工科太郎などを登録したテーブルを作成してみる。

```
sqlite> create table students(stdid INTEGER, name TEXT);
```

```
sqlite> insert into students values(12345678, '工科太郎');
```

```
sqlite> insert into students values(23456789, '工科花子');
```

```
sqlite> insert into students values(34567890, '工科次郎');
```

1 行目でテーブルを作成し、2,3,4 行目でテーブルにデータを登録している。

登録されているデータを取得していく。

```
sqlite> select * from students;
```

このように取得できる。

```
12345678|工科太郎
```

```
23456789|工科花子
```

```
34567890|工科次郎
```

登録されているデータをアップデートしていく。stdid の値を使いデータを指定する。

```
sqlite> update students set name="工科二郎" where stdid="34567890";
```

表示してみると、指定されたデータがアップデートされている。

```
sqlite> select * from students;
```

```
12345678|工科太郎
```

```
23456789|工科花子
```

```
34567890|工科二郎
```

データを削除する。stdid の値を使いデータを指定する。

```
sqlite> delete from students where stdid="34567890";
```

表示してみると、指定されたデータが削除されている。

```
sqlite> select * from students;
```

```
12345678|工科太郎
```

```
23456789|工科花子
```

接続中のデータベースを確認する。以下のコマンドを入力することで確認できる。

```
sqlite> .databases
```

```
main: /var/www/html/test/test.db r/w
```

テーブルの一覧を表示する。以下のコマンドを入力し、確認する。

```
sqlite> .tables
```

```
students
```

PHP から SQLite へ接続してデータベースを利用する

php.ini の設定を行う。「/etc/php/8.1/apache2/php.ini」の「;extension=pdo_sqlite」となっている部分のセミコロンを削除する。

```
$ gedit /etc/php/8.1/apache2/php.ini
```

936 行目あたりが次のようになっている

```
;extension=pdo_mysql
```

```
;extension=pdo_oci
```

```
;extension=pdo_odbc
```

```
;extension=pdo_pgsql
```

```
;extension=pdo_sqlite
```

```
;extension=pgsql
```

```
;extension=shmop
```

940 行目のセミコロンを削除し、次のようにする

```
extension=pdo_mysql
```

```
extension=pdo_oci
```

```
extension=pdo_odbc
```

```
;extension=pdo_pgsql
extension=pdo_sqlite
;extension=pgsql
;extension=shmop
```

さらに、「/etc/php/8.1/apache2/php.ini」の「;extension=sqlite3」となっている部分のセミコロンを削除する。948 行目あたりが次のようになっている。

```
;extension=soap
;extension=sockets
;extension=sodium
;extension=sqlite3
;extension=tidy
;extension=xsl
```

951 行目のセミコロンを削除し、次のようにする。

```
;extension=soap
;extension=sockets
;extension=sodium
extension=sqlite3
;extension=tidy
;extension=xsl
```

設定を変更したら、以下のコマンドを入力し Apache2 を再起動する。

```
$ sudo systemctl reload apache2
```

複数ある php.ini はどれが現在の使用されている設定ファイルなのかを確認していく。PHP の動作確認でアクセスした <http://192.168.64.7/test/info.php> の「Loaded Configuration File」に「/etc/php/8.1/apache2/php.ini」と書いてある。よって、現在使用されている php.ini は/etc/php/8.1/apache2/php.ini であることがわかる。

php-sqlite3 をインストールする。以下のコマンドを入力し、インストールする。

```
$ sudo apt install php-sqlite3
```

この操作後に追加で 154 kB のディスク容量が消費されます。

続行しますか? [Y/n]

このように聞かれたら Y を入力し、続行する。すると、php-sqlite3 がインストールされる。

以下のコマンドを入力し、Apache2 を再起動させ、インストールを完了させる。

```
$ sudo systemctl reload apache2
```

データベースを使用する PHP のサンプルを動かしてみる

授業ページにある dbtest.php を動かしていく。先に作成した「/var/www/html/test/test.db」を使用するため、このデータベースファイルと同じディレクトリに dbtest.php を設置する。よって dbtest.php のパスは「/var/www/html/test/dbtest.php」となる。

設置が完了したのち、Web ブラウザから dbtest.php にアクセスする。以下のように表示されていれば、正常に動作していることが確認できる。

```
array(4) { [0]=> int(12345678) ["stid"]=> int(12345678) [1]=> string(12) "工科太郎"
["name"]=> string(12) "工科太郎" }
array(4) { [0]=> int(23456789) ["stid"]=> int(23456789) [1]=> string(12) "工科花子"
["name"]=> string(12) "工科花子" }
```

セキュリティの観点から、パーミッションを変更する。所有者を www-data にし、この php ファイルは書き込まれると、正常に動作しない可能性があるため、所有者にのみ読み込み権限を与える。グループ、その他には権限を与えない。以下のコマンドで変更していく。

```
$ sudo chown www-data:www-data /var/www/html/test/dbtest.php
$ sudo chmod 400 /var/www/html/test/dbtest.php
```

変更が完了したら再度 dbtest.php にアクセスする。変更前と同じように表示されれば正常に変更が完了していることがわかる。

PDO を使用してデータベースを使用する PHP のサンプルを動かしてみる

PDO(PHP Data Object)は PHP のデータ接続クラス。PDO は同じ命令で複数のデータベースに接続ができるようになるデータ接続クラスである。なお、PDO は PHP の標準機能であるため、追加でインストールする必要はない。

マルチバイト文字を使用できるようにするため、php-mbstring をインストールする。以下のコマンドを入力し、インストールする。

```
$ sudo apt install php-mbstring
```

この操作後に追加で 1,732 kB のディスク容量が消費されます。

続行しますか? [Y/n]

このように聞かれたら Y を入力し、続行する。すると php-mbstring がインストールされる。

授業ページにある dbtest2.php を動かしていく。先に作成した「/var/www/html/test/test.db」を使うので、このデータベースファイルと同じディレクトリに dbtest2.php を設置する。よって dbtest2.php のパスは、「/var/www/html/test/dbtest2.php」となる。設置が完了したのち、Web ブラウザから

dbtest2.php にアクセスする。以下のように表示されていれば、正常に動作していることが確認できる。

```
12345678 工科太郎
```

所有者、グループ、パーミッションの設定を行う。Web サイトからアクセスするユーザは www-data であるため、所有者とグループは www-data にする。また、パーミッションは所有者のみが読み込み権限を持っていればよい。以下のコマンドで入力し、設定を適応させる。

```
$ sudo chown www-data:www-data /var/www/html/test/dbtest2.php
$ sudo chmod 400 /var/www/html/test/dbtest2.php
```

PHP でデータベースに登録された項目を色々と表示する

先で触れた dbtest2.php を修正し、「工科太郎」の名前だけ表示されるようにしてみる。dbdata2.php の 8 行目を次のように修正する。

```
$sql = "select name from students where stid = :stid";
```

任意の項目を foreach で表示する。これで「工科太郎」の名前のみ表示されるようになった。

次のようになるように表示する。

```
12345678
```

```
工科太郎
```

dbtest.php の 8 行目、16 行目、35 行目、36 行目を修正する。8 行目を次のように修正する。

```
$sql = "select * from students where stid = :stid";
```

16 行目を次のように修正する。

```
$rec = $stmt->fetchAll()
```

35 行目、36 行目を次のように修正する。

```
<?php foreach ($rec as list($stid_item, $name_item)):?>
```

```
    <?php echo $stid_item . "<br>" . $name_item;?>
```

これで任意の項目に対し、id と名前が改行され表示されるようになった。

PHP エラーが見えなくて困る場合

PHP がうまく動かない場合のエラー内容の確認方法について述べていく。先頭部分に次のように書いておくと、正常に動作した場合のみを除き、エラーがブラウザに表示されるようになる。

```
<?php
ini_set('display_errors', 1);
ini_set('error_reporting', E_ALL);
```

加点項目

以下のように表示されるよう、データベースとそのデータを用いて表示する PHP ファイルの作成を行う。

猿	バナナ
馬	にんじん
アルパカ	タピオカミルクティー

まずは、データベースの作成を行う。新たに test.db の中に table を作成し、データの登録を行う。以下のコマンドを入力し、テーブルを作成する。

```
$ sqlite3
sqlite> .open /var/www/html/test/test.db
sqlite> create table animals(name TEXT, food TEXT);
sqlite> insert into animals values('猿', 'バナナ');
sqlite> insert into animals values('馬', 'にんじん');
sqlite> insert into animals values('アルパカ', 'タピオカミルクティー');
sqlite> select * from animals
猿|バナナ
馬|にんじん
アルパカ|タピオカミルクティー
```

これでテーブルの作成が完了した。

次に表示するための PHP ファイルである test2_db.php ファイルを作成していく。授業サイトにある修正後の dbtest2.php を参考にし、test2_db.php ファイルを作成していく。以下の内容を test2_db.php に書き込んでいく。

```

<?php
ini_set('display_errors', 1);
ini_set('error_reporting', E_ALL);

$dsn = 'sqlite:/var/www/html/test/test.db';
try {
    . $pdo = new PDO($dsn);
    . $sql = "select * from animals";
    $stmt = $pdo->prepare($sql);
    $stmt -> execute();
    $rec = $stmt->fetchAll();
} catch(PDOException $e) {
    exit(mb_convert_encoding($e->getMessage(), 'UTF-8', 'SJIS-win'));
}

function printItem($str)
{
    return htmlspecialchars($str, ENT_QUOTES, 'UTF-8');
}

?>

<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<title>animals</title>
</head>
<body>
<table border="1">
<?php foreach ($rec as list($name_item, $food_item)):?>
    <tr>
        <?php echo "<th>" . $name_item . "</th> <td>" . $food_item . "</td>";?>
    </tr>
<?php endforeach; ?>
</table>
</body>
</html>

```

上記のように、<table>タグを用いて、表作成している。<table>タグの中の border 属性を border=1 で設定することで、二重線での格子で表示することができる。<tr>タグは行を表すため、データベースの 1 行を読み込み表示させる前後で囲う。<th>タグは、行名に当たるタグであるため、太字で表示されている。今回は動物名を太字にする指示であるため、name 要素を<th>タグで囲う。<td>タグは列の要素にあたる。今回は指示を満たすため、food 要素を<td>で囲う。

http://192.168.64.7/test/test2_db.php にアクセスし、動作確認を行う。指示通り表示されていれば問題ない。

所有者、グループ、パーミッション設定を行なっていく。Web ブラウザからアクセスするユーザは www-data である。よって所有者、グループ共に www-data にする必要がある。以下のコマンドで所有者、グループを変更していく。

```
$ sudo chown www-data:www-data /var/www/html/test/test2_db.php
```

この test2_db.php ファイルは所有者にのみ読み込み権限を与えておけば良い。以下のコマンドで権限を変更していく。

```
$ sudo chmod 400 /var/www/html/test/test2_db.php
```

再度、http://192.168.64.7/test/test2_db.php にアクセスし、先ほどと同じく、指示通り表示されていれば問題なく変更が行えている。